



软件工程师捉虫系列



DeBUGGING JAVA

Java

程序调试

买手用册

[美] Will David Mitchell 著
裘 岚 译

311-62

Mc
Graw
Hill

McGraw-Hill
<http://www.mhhe.com>



电子工业出版社
Publishing House of Electronics Industry
URL: <http://www.phei.com.cn>



Java 程序调试 实用手册

[美] Will David Mitchell 著

袁岚译

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 提 要

在开发应用程序时,最令你、你的主管和你的客户感到困扰的是无处不在,难以彻底消灭的漏洞,由于漏洞所造成的危害不胜枚举。如果你需要掌握开发无漏洞代码的思想、理论、技术和方法,那么请你认真阅读本书!本书是所有软件工程师的必读书籍,也可作为大专院校计算机专业师生的参考资料。

Will David Mitchell : DeBUGGING JAVA ,first Edition

ISBN 0-07-2125162-4

Copyright © 2000 by the McGraw-Hill Companies, Inc.

Authorized translation from the English language edition published by McGraw-Hill Inc.

All rights reserved.

本书中文简体字版由电子出版社和美国麦格劳-希尔国际公司合作出版。未经出版者书面许可,不得以任何方式复制或抄袭本书的任何部分。

版权所有,翻印必究。

图书在版编目(CIP)数据

Java 程序调试实用手册/(美)米查尔(Mitchell, W. D.)著;裘岚译.

—北京:电子工业出版社,2000.10

(软件工程师捉虫系列)

Debugging Java: Troubleshooting for Programmers

ISBN 7-5053-6286-0

I. J... II. ①米...②裘... III. JAVA 语言-程序设计-手册, IV. TP. 312-62

中国版本图书馆 CIP 数据核字(2000)第 72301 号

从 书 名: 软件工程师捉虫系列

书 名: **Java 程序调试实用手册**

原 书 名: **DeBUGGING JAVA**

著 者: [美] Will David Mitchell

译 者: 裘 岚 等

责任编辑: 寇国华

印 刷 者: 北京天竺颖华印刷厂

出版发行: 电子工业出版社 URL: <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 24.5 字数: 512 千字

版 次: 2000 年 10 月第 1 版 2000 年 10 月第 1 次印刷

定 价: 45.00 元

书 号: ISBN 7-5053-6286-0/TP • 3392

著作权合同登记号 图字: 01-2000-3017

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换。

若书店售缺,请与本社发行部联系调换。电话 68279077

译者的话

这是一本有极高参考价值的书，这是一本值得计算机行业中每个人认真且多次阅读的好书。

在有关武侠的文艺作品中，无论是雄心义胆的壮士好汉，还是狡诈阴险的歹毒小人，哪一个不想成为武林高手而独步天下？然而要达到这个目标，十八般兵器样样精通是远远不够的，而必须练就他人不可问鼎的一身绝技！在软件开发领域，最高境界是开发无任何漏洞的代码。作为一个软件工程师，哪一个不想成为顶尖高手？然而要达到最高境界，并不是精通各种开发语言和开发工具就可以实现的，而必须要掌握开发无任何漏洞代码的思想、理论、技术和方法。

译者自觉才疏学浅，由于工作关系，一直在寻觅这方面的有关书籍，一次次地在书店里成堆的计算机图书中查找，可能是诚心未到，只能一次次地失望而归。当拿到本书原著浏览之后，我惊喜地发现，踏破铁鞋无觅处，得来全不费工夫！

本书作者作为一名计算机专家，曾经担任过计算机科学专业的大学教授，有着丰富的教学经验，善于使用浅显易懂的文字和轻松的样例说明现象背后的本质，并经常用自身的经历作为样例讲解。

本书围绕如何调试 Java 程序这一主题展开，用大量篇幅讲解了开发无漏洞代码的思想、理论、技术及方法。可以肯定地说，本书的内容不仅适用于 Java 程序开发人员，而且适用于所有软件开发人员，例如书中提到的编程 24 条法规。对于项目开发的管理人员，系统设计师等，也可从本书中获益匪浅。利用书中的知识组织或指导项目开发，可以大大提高工作效率，大大缩短开发周期，大大节省开发经费。

在翻译过程中，我曾经多次击案叫绝，深深地为作者独到而明确的见解，广博而精深的学识，丰富而成熟的经验，精辟而深入的分析所折服。译稿杀青之时，欣喜之余，我唯一祈盼的就是这本书能够和广大读者尽快见面，以使读者能够为早一天阅读到这本极其难得的好书而惊喜。

由于时间比较紧，工作量大，加上译者水平有限，书中难免有疏漏和失误之处，还请读者指正。参加本书翻译工作的还有张翔、白雪、张准野、段来盛、韩珊、孙春来、宋黎松、官章全以及胡存生等同志，在此一并感谢。

译者
2000 年 9 月

前言

只有上帝才知道有多少人为这本书做出了巨大的贡献，要将他们的名字全部列出来的话，可能还需要再写一本书才行。我得到了许多珍贵的祝福。

我的爱妻 Carol 和爱女 Christie 的水平远远胜过一般的作家，她们和我的小儿儿子 Andrew 一起在这本书上耗费了大量时间和精力，他们帮助我校对章节内容，出其不意地为我买来快餐，并且给予我巨大的鼓励。对于他们所做出的所有这些牺牲，我简直无法找到恰当的词汇来表达我的感谢和爱。有许许多多的人为我提供了样例程序，并表现出解除寻找漏洞苦恼的幽默。这些好朋友包括 Galen、Barb、Robert、Frank、Lonnie、Bill、Lou、Ted、Gary、Steve、Sue、Evelyn、Clarke、Joe、Sam、Roy、Sandy、Dennis、Russ、Don，以及许多叫 Mike、John 和 David 的人，你将在本书的各个角落看到他们的贡献，在这里我要感谢他们中的每一个人。

向我教授计算机科学那些人同样是贡献者，我非常希望自己可以亲自向每一个人道谢。特别是其中四位专家：Marilyn Mantel-Guss、Hsing Liu、Stan Wileman 和 Matthew Payne。

之所以到最后才提到以下的人，只是为了强调他们所做出的贡献。现在已经提升为主编的助理编辑 Wendy Rinaldi 是这类项目完美的提倡者，当初也正是她找到我来写这本书。她浑身洋溢的热情和奇妙的念头促成了这个系列的书籍，而她那能干的秘书 Monika Faltiss 负责整理所有日程安排的细节。我真的是非常幸运，能够找到杰出的 Mark Karmendy 作为项目编辑。当我看到本书最终的封面时，我认为这是我曾经见过的两本最棒的书籍封面之一。William Voss 和 Dodie Shoemaker 设计了封面。同样我还要真诚地感谢构思调试系列的作家 Chris Pappas 和 William Murray。

内容简介

计算机科学的哪个方面最令你、你的主管和你的客户(内部或者外部)感到困扰？是漏洞吗？那么你并不孤单。地球上最为强大的市场商人总是能够发现最能导致人们困扰的问题，之后就致力于减轻痛苦。由于同样的原因，世界上最令人尊敬的程序员也是通过制造无漏洞无烦恼的代码成为英雄的。这些绝好的技术中有许多都与直觉相矛盾，但是本书为你说明这一切，帮助你如何：

- 从一开始就预防漏洞的产生。
- 编写无漏洞的代码。
- 找到并清除那些遗漏的漏洞。
- 创建一个较好的漏洞捕获程序。

- 编写完美的错误消息。
- 使用强大的自动工具从程序中清除漏洞。
- 让你的用户更爱你。

漏洞之所以造成困扰是因为其代价非常昂贵。想象一下千年虫造成的数百亿损失，或者重新替换由于错误在火星上撞毁的空间探测器的费用。尽管在数学上是不可能测试所有的漏洞的，但是防止它们在你的代码中滋生却是切实可行的。本书揭示了程序员实现这一目标的秘诀。尽管本书内容是有关 Java 及其相关产品的，但是作者的准则可以运用到任何一个项目中，甚至包括计算机科学之外的领域。本书提供了某些 Java 漏洞的源代码，以及消灭漏洞的妙方。

多数公司只测试自己销售的代码中近三分之一的部分！导致的结果就是在全世界的午餐桌上被所有人抱怨。Java 提供了一些自动处理的强大工具帮助调试程序。使用这些工具可以确保测试所有的代码。你会发现测试过程要比不久前测试的任何代码都彻底。本书说明了如何使用这些绝妙的自动工具。

说真的，你也可以拥有作者已经保持了二十年的座右铭，那就是“永远保证软件与规范说明的一致性。”

作者简介

Will David Mitchell 从 1973 年开始从事计算机方面的工作。在内布拉斯加大学教授计算机科学课程期间，他的研究就表明首先学习调试技术的程序员可以更快掌握计算机语言。因此，他在教授的班级中从第二个星期开始就强调学习调试技术。当时学校中任何一个班级都没有在标准测验中取得过 73% 的通过率，而 Mitchell 的班级却始终将通过率保持在 80-82% 的范围之内。这之间唯一的区别就是他较早强调了调试技术。在本书中，作者揭示了帮助你尽早成为 Java 专家的秘诀。

Mitchell 从 1971 年开始为专业杂志写文章，到目前为止已经发表了一千多本技术参考书、文章和论文。最近几年出版了五本高科技小说。作者的热门站点网址为 <http://weblications.net>。除了是一名作家和计算机科学家之外，Mitchell 还是一位独立的计算机顾问、喷气机飞行员、数学家、艺术家和音乐家。他住在内布拉斯加州的奥马哈附近。

目 录

第一部分 从编写没有漏洞的代码开始

第 1 章 完全没有漏洞是不可能的.....	3
1.1 证据.....	4
1.2 调试或者测试都无法找到所有漏洞.....	5
1.3 这样将变得更糟.....	8
1.4 开始就必须去除代码漏洞.....	8
第 2 章 使用 Hatching 预防 Java 漏洞.....	11
2.1 开发哲学.....	12
2.1.1 左脑 $\rightleftharpoons$ 右脑.....	12
2.1.2 如何更加富有创造力.....	14
2.1.3 如何更好地组织.....	17
2.1.4 程序员创建而测试员破坏.....	17
2.2 首先编写文档.....	18
2.2.1 首先是用户手册.....	18
2.2.2 强有力的结束工作.....	19
2.2.3 用户手册成为编程规范.....	19
2.3 学着喜欢 Javadoc.....	20
2.4 危险元素在安全元素之前.....	22
2.4.1 尽早处理例外.....	22
2.4.2 避免限期压力.....	23
第 3 章 设置 Java 漏洞中断.....	25
3.1 指导代码.....	26
3.1.1 代码工具.....	26
3.1.2 Java 例外的细节.....	28

3.1.3	例外提高了艺术的境界	30
3.1.4	处理例外的编码	36
3.1.5	嵌套使用 try 模块	38
3.2	throw 模块	38
3.2.1	throws 关键词	39
3.3	隐藏在幻象漏洞之后的漏洞	41
3.3.1	练习生成漏洞	42
3.4	自动记录结果	42
第 4 章	千万不要错过另一个限期！危险因子分析	45
4.1	使用危险因子分析(RFA)	47
4.1.1	如何开始 RFA	47
4.1.2	如何使用 RFA	47
4.1.3	为什么使用 RFA	49
4.2	为什么 RFA 对于调试 Java 非常重要？	49
第 5 章	编写代码避开漏洞	51
5.1	通用文字处理器	52
5.1.1	使用 Microsoft Word	55
5.2	使用最好的 Java 编辑器	65
5.3	编辑窍门	66
5.3.1	使笔误自我显露	69
5.3.2	扩展拷贝/粘贴缓冲区	69
5.3.3	使用自动更正功能清除错误和保存输入	72
5.4	练习拼写检查器	74
5.5	按照先头后尾再中间的顺序书写	75
5.6	QQQ 书签	76
5.7	3X5 的打孔卡片	77
5.8	使用已知的 Java 子集	79
5.9	先注释后代码	80
5.10	牢记语言之间的区别	81
5.10.1	Java 和 C/C++	81
5.10.2	Java 和 Visual Basic(VB)之间的主要区别	85
5.11	集成开发环境(IDE)	89

5.11.1	JBuilder	90
5.11.2	JDK Commander	92
5.11.3	Mojo	92
5.11.4	VisualCafe	92
5.12	漏洞类别	94

第二部分 清除 Java 漏洞

第 6 章	漏洞类别	97
-------	------------	----

6.1	设计漏洞	98
6.1.1	条件总是以 2 的幂数成对出现	98
6.2	语法漏洞	100
6.2.1	代码生成器	100
6.2.2	代码生成器的特性	103
6.2.3	类似 Lint 的检验程序	105
6.3	逻辑漏洞	113
6.3.1	逻辑性实际错误	113
6.4	解决神秘之处	118
6.5	数学漏洞	119
6.5.1	接近边界值的数学问题	120
6.5.2	布尔变量	121
6.5.3	不常用的操作符: 移位	124
6.6	罕见的漏洞	129
6.6.1	数据导致的漏洞	130
6.7	副作用漏洞	130
6.8	优化引起的漏洞	131
6.9	假冒的漏洞	132

第 7 章	心理训练	135
-------	------------	-----

7.1	如何保持思维的一贯性	136
7.1.1	使用纯粹的个人习惯	136
7.2	不要混合使用深度搜索和广度搜索	142
7.3	何时调试	144
7.4	环境	145

第 8 章	Debugger 的可怕威力	147
8.1	免费的 JavaDebugger(JDB)	149
8.1.1	安装	149
8.1.2	简介	149
8.1.3	命令参考	150
8.2	第三方 Debugger	151
8.2.1	Assure	152
8.2.2	JBuilder	158
8.2.3	JProbe	165
8.2.4	Visual Cafe	171
第 9 章	调试策略	183
9.1	集成最好资源	184
9.2	分解漏洞	184
9.2.1	开始修改	185
9.2.2	猎枪的方法	186
9.2.3	根据推论调试	186
9.2.4	二进制漏洞搜索	188
9.2.5	测试	191
9.3	卡住时问些问题	193
第 10 章	测试	195
10.1	定位后击垮漏洞	196
10.1.1	武装你的代码	196
10.1.2	条件编译	197
10.1.3	漏洞在何处?	197
10.1.4	黑盒测试	203
10.1.5	白盒测试	204
10.1.6	全逻辑测试	206
10.2	制造更好的苍蝇拍	206
10.2.1	Macro Recorder	207
10.2.2	Best Practices Analyzer	207
10.2.3	Static Coverage Analyzer	208

10.2.4	Dynamic Coverage Analyzer.....	208
10.2.5	Bug Tracker.....	208
10.2.6	Test Data Assistant.....	210
10.2.7	何时停止测试	210
10.2.8	播撒错误的种子	215
10.2.9	你需要第二台计算机	216
10.3	Java 的商业软件工具.....	219
第 11 章	线程化环境	223
11.1	回顾古老的并行算法.....	225
11.2	并行计算漏洞.....	227
11.2.1	Daemon 和 user 线程	227
11.2.2	Java 本身的防护	227
11.2.3	循环并行化.....	228
11.2.4	处理输入和输出.....	228
11.2.5	异步任务.....	230
11.2.6	定时程序.....	231
11.2.7	对时间敏感的线程漏洞	232
11.3	预防线程崩溃.....	235
11.3.1	使用循环锁解救.....	237
11.3.2	使用监督程序和信号量锁定	239
11.3.3	监督程序和信号量的详细内容	242
11.3.4	一些线程使用经验.....	243
11.3.5	性能问题.....	245
11.4	线程安全	245
11.5	预防措施.....	246
第 12 章	走开的人	247
12.1	用户如何查觉漏洞.....	248
12.1.1	让你的客户喜欢你	250
12.2	定义造成的漏洞混乱.....	252
12.3	还有什么可能出错?	253
12.4	组成完美错误信息的元素.....	253
12.4.1	使用用户的语言描述	253

12.4.2	不要过于简洁	254
12.4.3	小心选择词汇	254
12.4.4	确认是否拼写错误	254
12.4.5	道歉永远不会有害	255
12.4.6	最好完全解密	255
12.4.7	你的用户正处在接近恐慌的状态	255
12.4.8	错误消息必须可以缓和情绪	255
12.4.9	避免使用任何屈尊的语气	256
12.4.10	标准化步骤	256
12.4.11	按钮标题	258
12.5	格式化错误消息	258
12.6	错误消息的内容	259
12.6.1	发生了什么事?	259
12.6.2	为什么发生?	259
12.6.3	其后将发生什么现象?	260
12.6.4	现在用户可做什么?	260
12.6.5	将来用户能做什么?	261
12.6.6	现在用户从何处可以得到帮助?	261
12.6.7	用户如何才能帮助开发人员改善情况?	261
12.6.8	最近在用户的软件中发生过类似问题否?	262
12.6.9	用户应该如何向技术人员描述问题?	264
12.6.10	聊天室和帮助室	265
12.6.11	软件开发人员将为用户提供什么补偿?	265
12.6.12	问题发生时计算机的状态如何?	266
12.6.13	漏洞在客户端还是在服务器端?	266
12.6.14	打开了哪个数据库、表和字段?	266
12.6.15	哪个程序、哪个模块、哪种方法以及哪一行触发了错误?	266
12.6.16	当时哪个线程正在活动?	267
12.6.17	按照计算机支持的精度, 问题究竟在何时发生?	267
12.6.18	登录用户是哪一位?	267
12.6.19	屏幕或者报告中应该显示什么内容	268
12.7	永远按照规范保证自己的软件	268

第三部分 性能

第 13 章 使用最佳的测试策略	271
13.1 递增与模块测试	272
13.1.1 递增测试的优点	272
13.1.2 传统模块测试的优点	273
13.1.3 综合测试	273
13.2 从上至下测试与从下至上测试	273
13.2.1 从上至下测试	273
13.2.2 流程图为什么有缺陷	274
13.2.3 从下至上测试	275
13.2.4 协议	275
13.3 原理测试	275
13.4 测试流程图的空白处	276
13.5 自动测试程序	277
13.5.1 自动黑盒测试	281
13.5.2 自动白盒测试	281
13.5.3 自动回归测试	282
13.5.4 自动静态分析	283
13.5.5 自动覆盖分析	289
13.5.6 幻想和神话	292
13.6 清除漏洞所需的费用	293
13.7 其他种类的测试	293
13.8 还剩下多少漏洞?	294
附录 A 商业设计	297
A.1 附加项类库	298
A.2 人工智能	298
A.3 浏览器及浏览器工具	299
A.4 代码生成器	299
A.5 协作者	301
A.6 编译程序和解释程序	301
A.7 数据和网络数据	301

A.8	调试程序.....	305
A.9	文档编写器.....	306
A.10	电子商务.....	307
A.11	编辑器.....	308
A.12	图形开发.....	308
A.13	帮助文档编写器.....	310
A.14	IDE 和开发工具.....	310
A.15	安装与配置.....	313
A.16	国际化.....	314
A.17	Internet.....	315
A.18	Java Beans 和企业版的 Java Beans(EJB)	315
A.19	Java 虚拟机.....	318
A.20	JAR 自解压程序.....	319
A.21	制图程序.....	319
A.22	消息程序.....	319
A.23	建模, UML, 和 CASE 工具	320
A.24	Obfuscator 和优化程序	323
A.25	对象请求代理程序(ORBs)	324
A.26	Profiler.....	324
A.27	报表制作程序.....	324
A.28	安全.....	326
A.29	服务器和 Servlet	326
A.30	软件开发工具包.....	328
A.31	电子制表软件.....	328
A.32	测试工具和套件.....	329
A.33	跟踪器, 工程管理器.....	329
A.34	语音识别.....	330
A.35	Java 文字处理器.....	330
A.36	XML.....	330
附录 B	Java 资源	333
B.1	书籍	334
B.2	杂志	334
B.3	电子杂志	335

B.4	新闻组	336
B.5	Web 站点	336
B.6	培训	337
附录 C	计算机编程的 24 条法规	341
附录 D	Java 术语表	345
附录 E	Word 宏	355
E.1	书签和跳转	356
E.2	隐藏的文本	357
E.3	个人注释	359
E.4	绿色的关键字	359
E.5	编程帮助	374
E.6	更多的宏	376

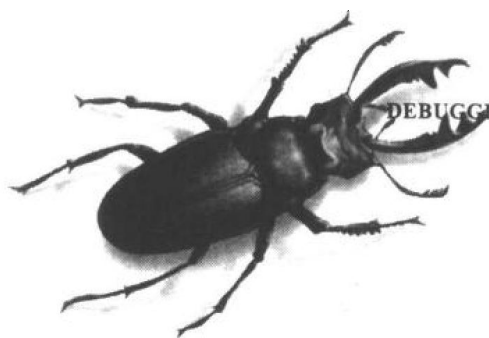
第一部分

DEBUGGINGDEBUGGINGDEBUGGINGDEBUGGING

从编写没有漏洞的代码开始

DEBUGGINGDEBUGGINGDEBUGGINGDEBUGGINGDEBUGGINGDEBUGGINGDEBUGGING

- 第 1 章 完全没有漏洞是不可能的
- 第 2 章 使用 Hatching 预防 Java 漏洞
- 第 3 章 设置 Java 漏洞中断
- 第 4 章 千万不要错过另一个限期！危险因子分析
- 第 5 章 编写代码避开漏洞



DEBUGGINGDEBUGGINGDEBUGGINGDEBUGGINGDEBUGGINGDEBUGGING

DEBUGGINGDEBUGGINGDEBUGGINGDEBUGGINGDEBUGGING

